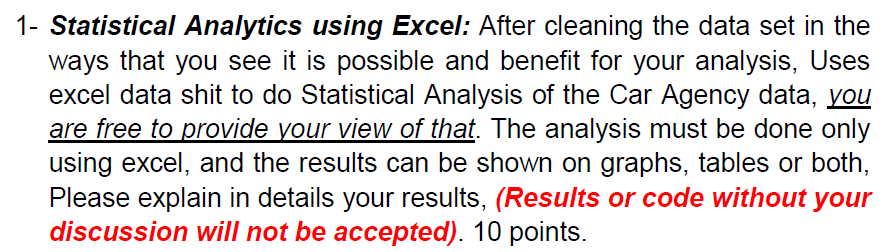
**Big Data and Analytics**

**Introduction**

The data downloaded from Kaggle (here is the link: <https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho?select=car+data.csv>). The purpose of the car dataset analysis is to create an online system to help managers of used car agencies to search for a car, by various different values and inputs. In the data set, there are 13 variables where 8 variables are quantitative and 5 are qualitative variables. The dataset consists of 8128 observations and missing values exist in the dataset.

**1. Statistical Analytics using Excel**

****

Cleaning Data in Excel:

In this section, we clean the dataset using MS excel. The variable name is “name” which consists of the car's name. There are 31 different car company names. For simplicity, we split names from the name variable and store it as a new variable as “brand”. This is done with the help of “text to columns”. The benefit of the step is to extract information easily such as information about brand is much easier to search or input as compared to model name. On the other hand, we can treat the “brand” variable as a categorical variable.

The units of “mileage” variable are given two types in the dataset that is kmpl and km/kg. So, we have to convert the values on the same units because different types of units disturb the accuracy of the model. For conversion of units from km/kg to kmpl, multiply kmpl to 1.4. Thus, the values show the same units in the mileage (kmpl) column.

Table 1: Counts and Missing Values

|  |  |  |
| --- | --- | --- |
| **Variable** | **Count** | **Missing Value** |
| Brand | 8128 | 0 |
| Year | 8128 | 0 |
| selling\_price | 8128 | 0 |
| km\_driven | 8128 | 0 |
| fuel | 8128 | 0 |
| seller\_type | 8128 | 0 |
| transmission | 8128 | 0 |
| owner | 8128 | 0 |
| mileage | 7907 | 221 |
| engine | 7907 | 221 |
| max\_power | 7913 | 215 |
| torque | 7906 | 222 |
| seats | 7907 | 221 |

The above tables show the count and missing values of the variables. Looking at the missing value column, if the value is zero means that there is no missing value in the dataset, on the other hand, if the value is non-zero means that there is missing value in the data set.

Handling missing value in the data set:

The Mileage variable has 221 missing values. The data type of the mileage variable is a continuous variable. According to the statistics, when the data type is continuous, you can fill the missing value with mean or average of mileage.

The engine (CC) has 221 missing values. The data type is numerical and the distribution of the data set is skewed because the value of the skewness value is 1.14 which means that the value of the skewness is greater than 1, the data are highly skewed. According to the statistics, when the data is skewed, you have to fill the value with the median. The median value of the engine (cc) is 1248. Thus, the missing value is filled by the median.

The variable “max power” (bhp) has 215 missing values. The data type of the max power variable is a continuous variable. So, when the data type is continuous, you can fill the missing value with mean or average of max power. Thus, the max power is filled by its mean.

The variable “seats” has 221 missing values. The data type is numerical and the distribution of the data set is skewed because the value of the skewness value is 1.97 which means that the value of the skewness is greater than 1, the data are highly skewed. So, when the data is skewed, you have to fill the value with the median. The median value of the seats is 5.. Thus, the missing value is filled by its median.

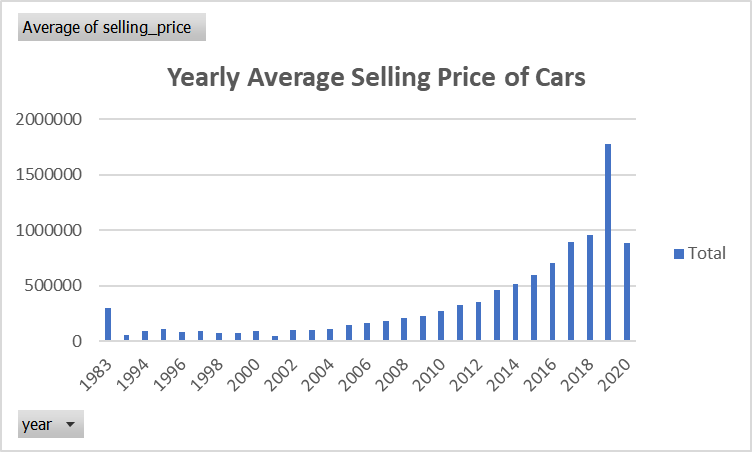
After, filling all missing values in the dataset. We convert categorical variables into numerical variables because it gives better outcomes. The fuel, seller\_type, transmission, and owner is converted to a dummy variable.

Table 2: Data Description (Conversion of Categorical to Dummy variable) are as follows:

|  |  |
| --- | --- |
| **Variable and description** | **Value** |
| fuel | Diesel = 0  Petrol = 1  LPG = 2  CNG = 3 |
| seller\_type | Individual = 0  Dealer = 1  Trustmark Dealer =2 |
| transmission | Manual = 0  Automatic = 1 |
| owner | First Owner = 0  Second Owner = 1  Third Owner = 2  Test Drive Car = 3  Fourth & Above Owner = 4 |

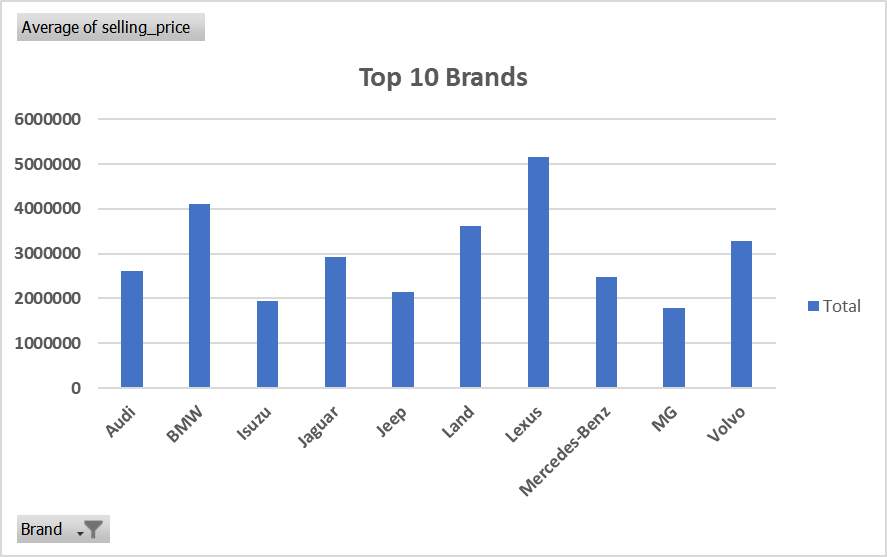
Graphical Representation:

1. Yearly Average Selling Price of Cars



The above graph shows the yearly average selling price of all cars. As the graph shows, average sale prices of the cars were lowest between 1994 to 2004 and it was around 90,000. Gradually, it will increase after 2010 and sales prices were at their peak in the year of 2018 and reached 957770. We may conclude that the demand for cars increases gradually, so the prices of the cars increase.

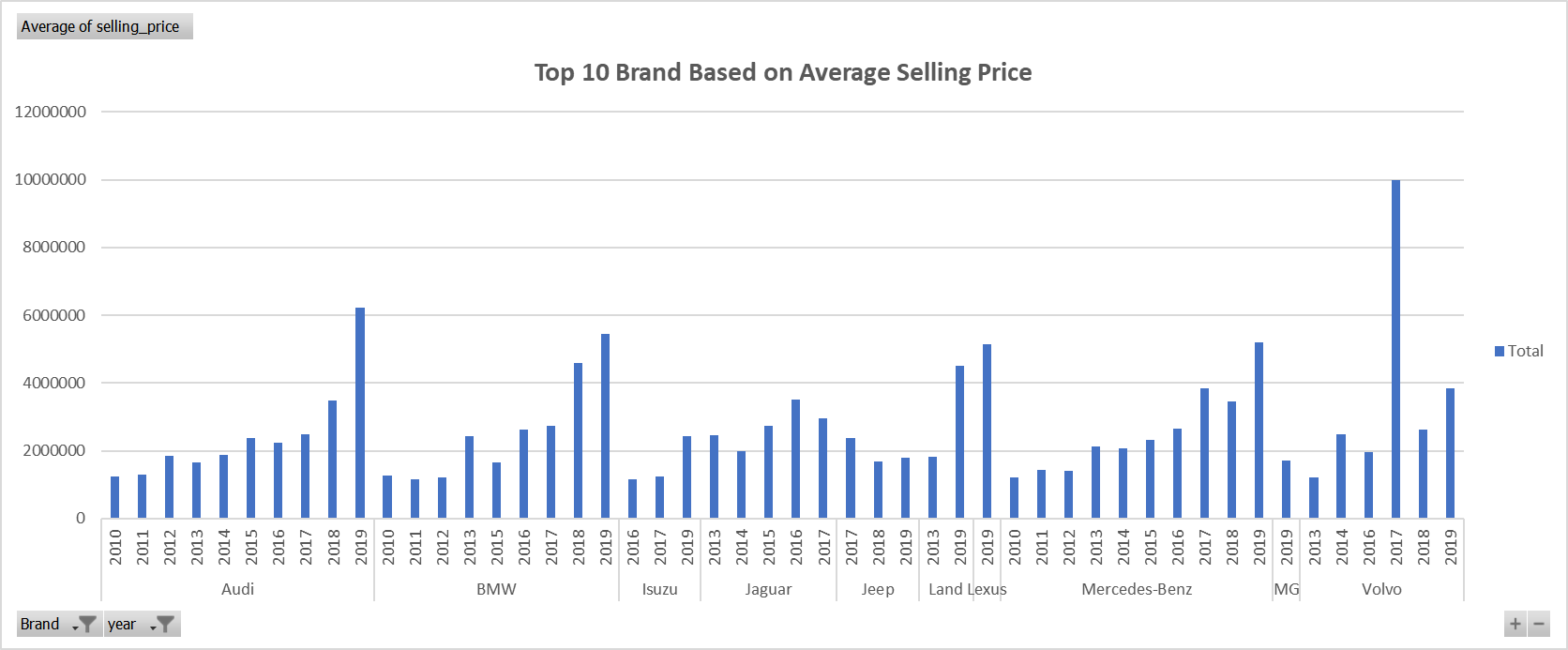
2. Top 10 Car Brands



The above graph shows the top 10 brands average selling price. The Lexus had the highest average selling price as compared to the other brands.

On the other hand, the MG had the lowest average selling price as compared to the other brands.

3. Top 10 Brand Based on Average Selling Price (Between 2010 to 2019).

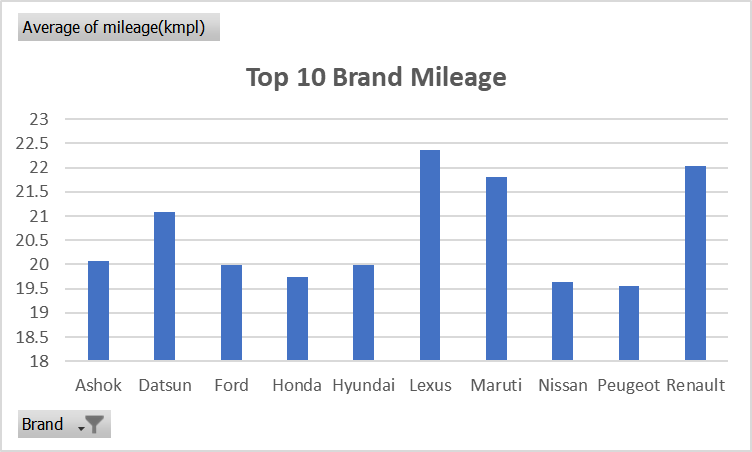


The above graph displays the top 10 brands based on the average selling price in between 2010 to 2019. These are the top 10 brands of cars: Audi, BMW, Isuzu, Jaguar, Jeep, Land, Lexus, Mercedes-Benz, MG, and Volvo.

In the year of 2018, BMW (4585000) had the highest selling price as compared to AUDI (3475000). On the other hand, in the year of 2019, AUDI (6223000) had the highest selling price as compared to BMW (5453125). Therefore, both are competitors of each other.

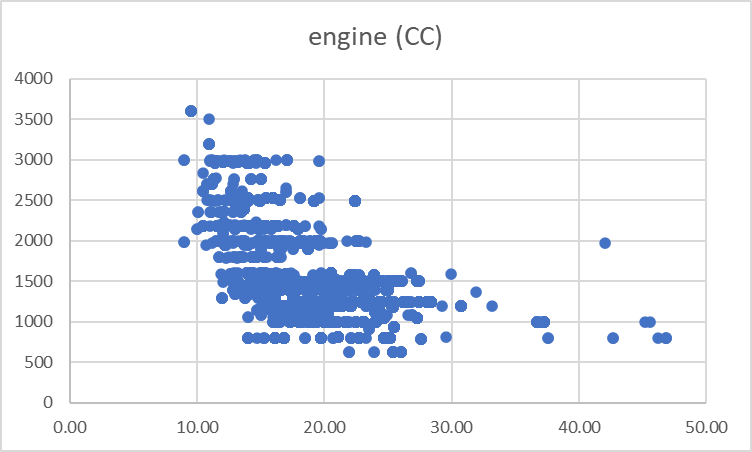
If we compare the recent year's price, that is 2019, then AUDI's selling price is highest among 10 brands. In addition, from 2010 to 2019, Volvo had the highest average selling price in 2017. Furthermore, Isuzu had the lowest average selling price as compared to other brands. Thus, Isuzu is a less competitive brand compared to other brands.

4. Top 10 Brand Mileage



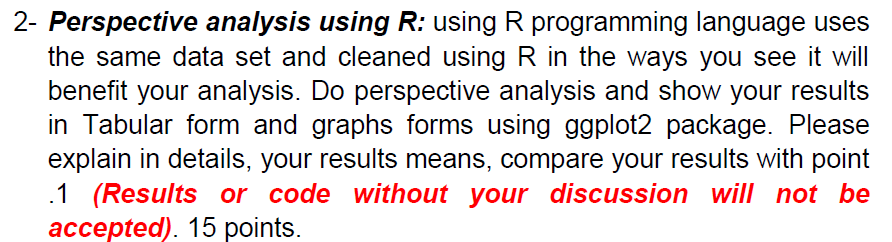
The above diagram displays the top 10 brand mileage. It can be easily seen that, on an average Lexus had the highest mileage among the 10 brands. On the other hand, on an average the Renault had the second highest mileage. However, this is not the real case scenario, the result is based on the dataset.

5. Mileage vs Engine



The above diagram shows the scatter plot between the engine (CC) and mileage of the cars. The scatter plot is more appropriate for these variables because its data points are continuous. It shows a negative relationship between the engine (cc) and mileage of the cars for the reason that plot points seem to form a line going down from left to right. In general, the larger the engine (CC's), the lower mileage it will achieve. That would hold true for petrol and diesel engines.

Answer 2:

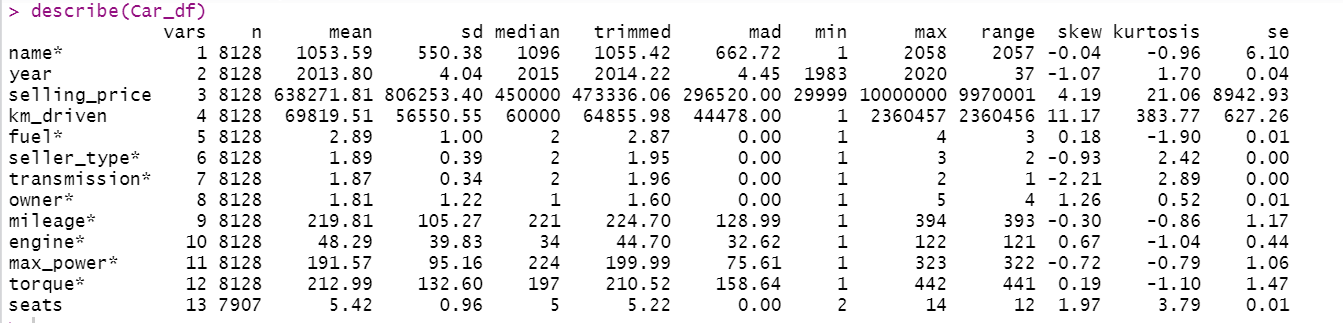


Data preprocessing:

In this section, we clean the dataset using R programming. The variable name is “name” which consists of the car's name. For simplicity, we split names from the name variable and store it as a new variable as “Brand”. This is done with the help of “strsplit () function”. The benefit of the step is to extract information easily such as information about brand is much easier to search or input as compared to model name. On the other hand, we can treat the “brand” variable as a categorical variable.

The units of “mileage” variable are given two types in the dataset that is kmpl and km/kg. So, we have to convert the values on the same units because different types of units disturb the accuracy of the model. For conversion of units from km/kg to kmpl, multiply kmpl to 1.4. So, firstly, we create a variable “Status” and using the if else condition, if the unit starts from “g” then assign 1 and else it assign 0. After that, we multiply 1 to 1.4 and rest values are the same as it is. Thus, the values show the same units in the mileage (kmpl) column.

Table 1.1: Summary Statistics



Handling missing value in the data set using R:

The Mileage variable has 221 missing values. The data type of the mileage variable is a continuous variable. According to the statistics, when the data type is continuous, you can fill the missing value with mean or average of mileage.

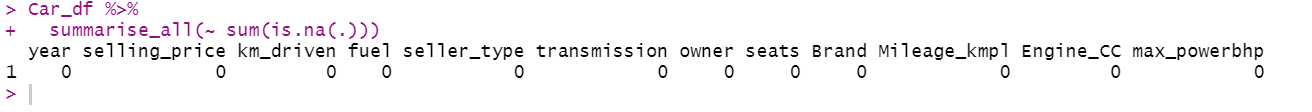
The engine (CC) has 221 missing values. The data type is numerical and the distribution of the data set is skewed because the value of the skewness value is 1.14 which means that the value of the skewness is greater than 1, the data are highly skewed. According to the statistics, when the data is skewed, you have to fill the value with the median. The median value of the engine (cc) is 1248. Thus, the missing value is filled by the median.

The variable “max power” (bhp) has 215 missing values. The data type of the max power variable is a continuous variable. So, when the data type is continuous, you can fill the missing value with mean or average of max power. Thus, the max power is filled by its mean.

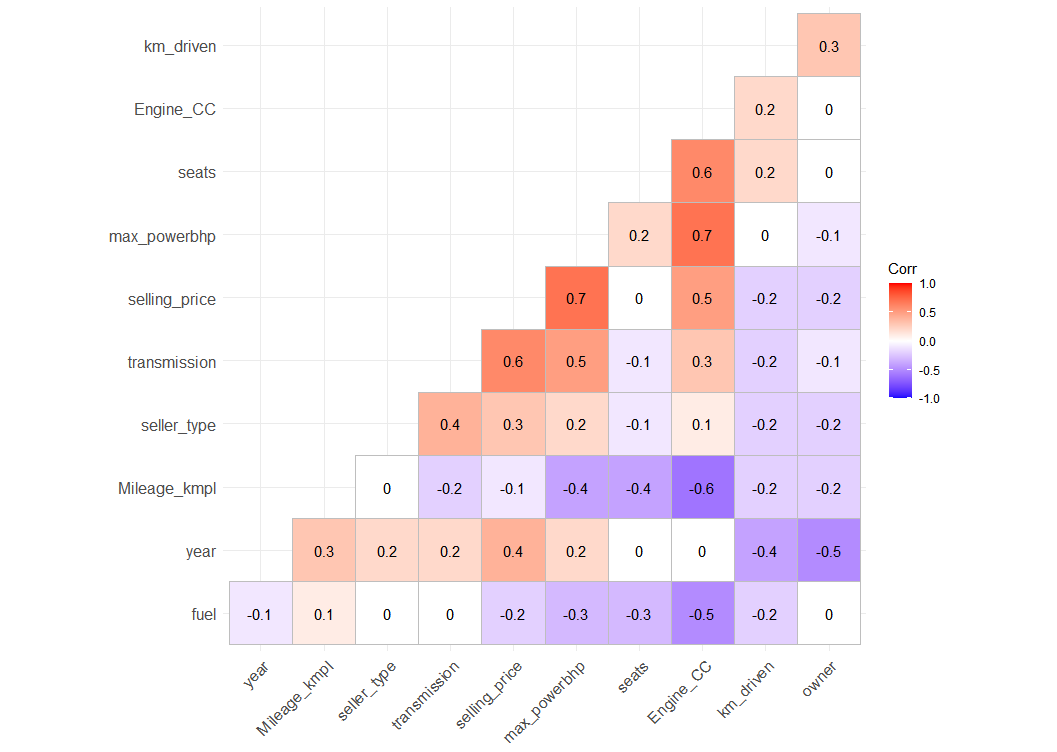
The variable “seats” has 221 missing values. The data type is numerical and the distribution of the data set is skewed because the value of the skewness value is 1.97 which means that the value of the skewness is greater than 1, the data are highly skewed. So, when the data is skewed, you have to fill the value with the median. The median value of the seats is 5.. Thus, the missing value is filled by its median.

After, filling all missing values in the dataset. We convert categorical variables into numerical variables because it gives better outcomes. The fuel, seller\_type, transmission, and owner is converted to a dummy variable (same as the table 2).

After implementing all the above processes, no missing values exist in the dataset (see below table).



# Check Correlation

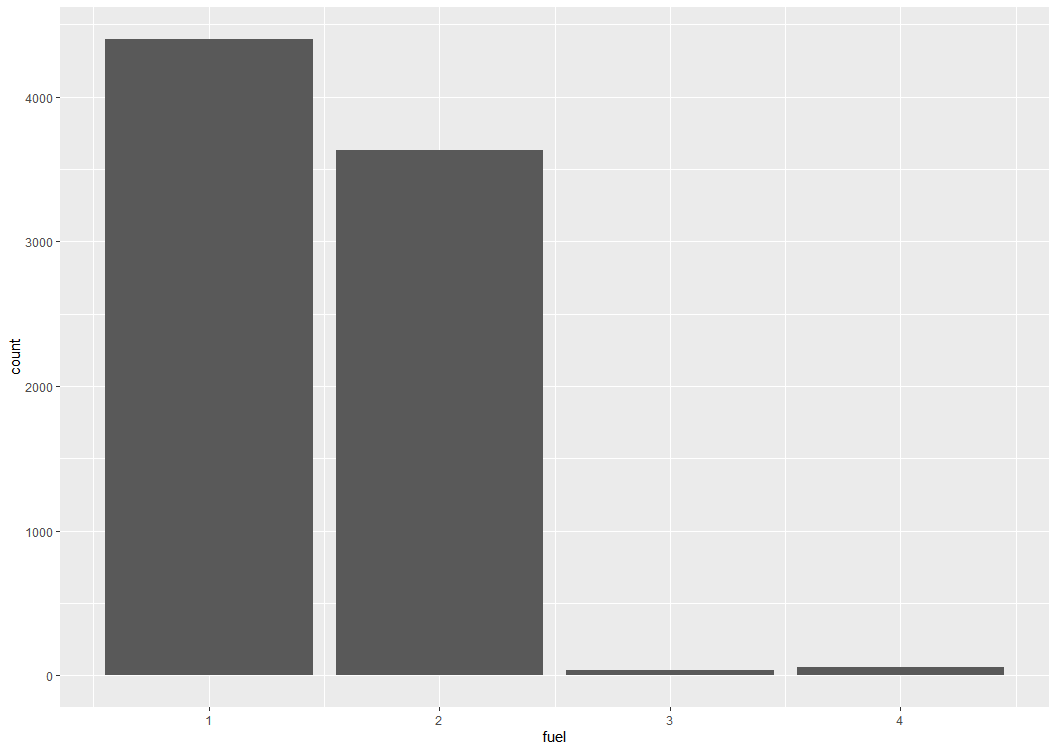


* ‘Engine cc’ and ‘max power bhp’ are highly correlated.
* Selling price and 'year' are moderately correlated.
* 'km Driven' and selling price show a negative correlation.
* Engine and Mileage shows a moderate negative correlation.
* 'owner' and selling price show a negative correlation.

**Graphical Representation of the cleaned data set using R:**

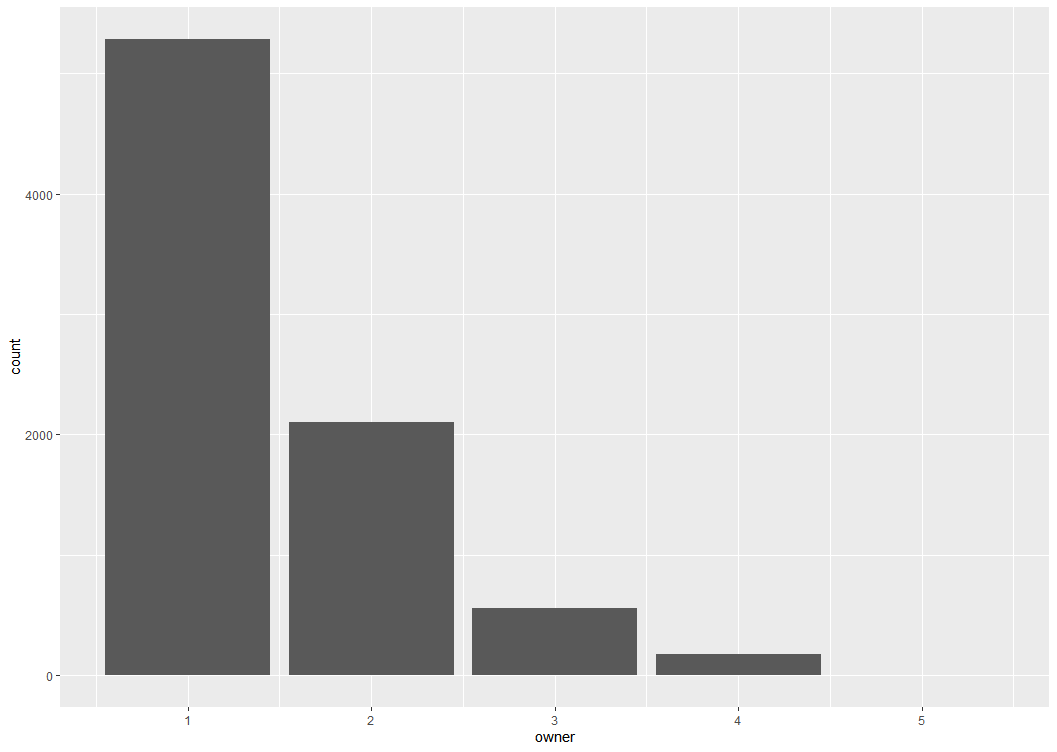
Explore Categorical Variable

1. Fuel



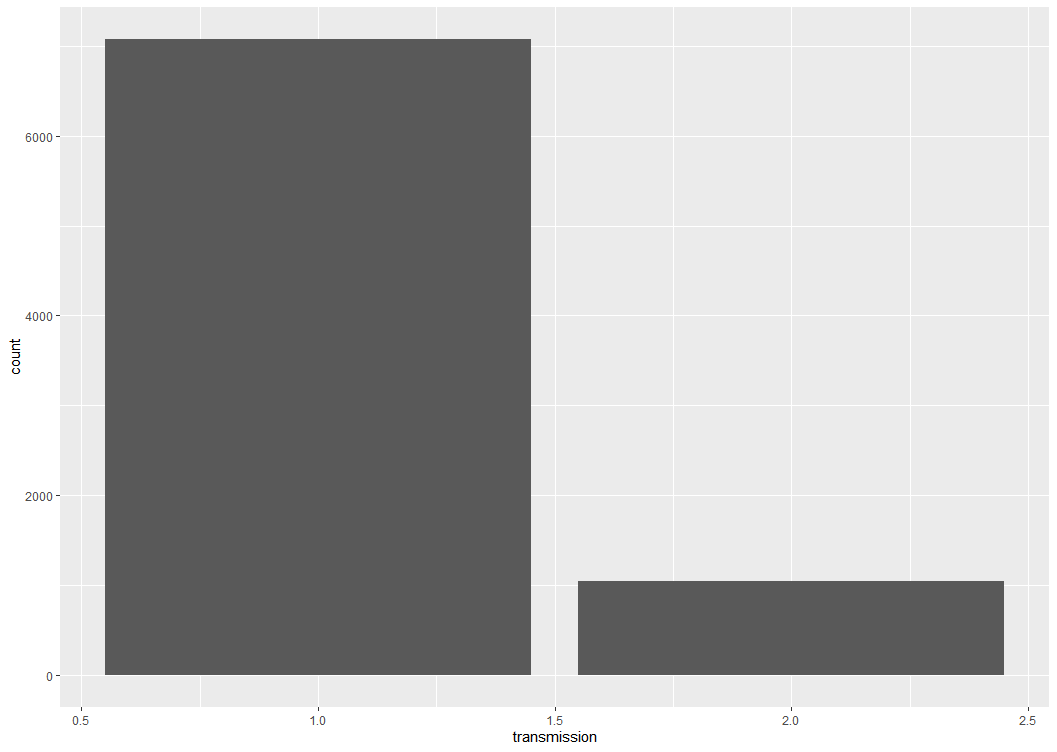
There are 4 categories of fuel that are Diesel, Petrol, CNG, and LPG. As the bar graph shows , most cars had a diesel engine while the least cars had CNG. The second highest cars had a petrol engine.

2. Owner



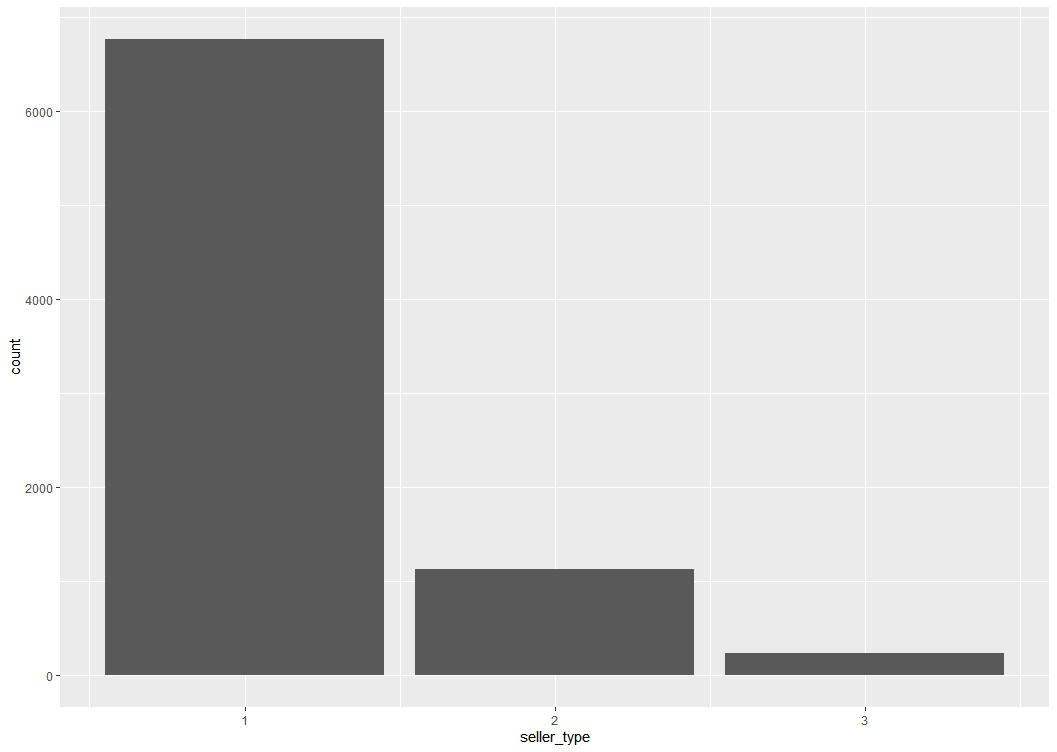
Most owners prefer to sell first owner cars as compared to second owners. There are less or no owners who prefer to sell test drive cars.

3. Transmission



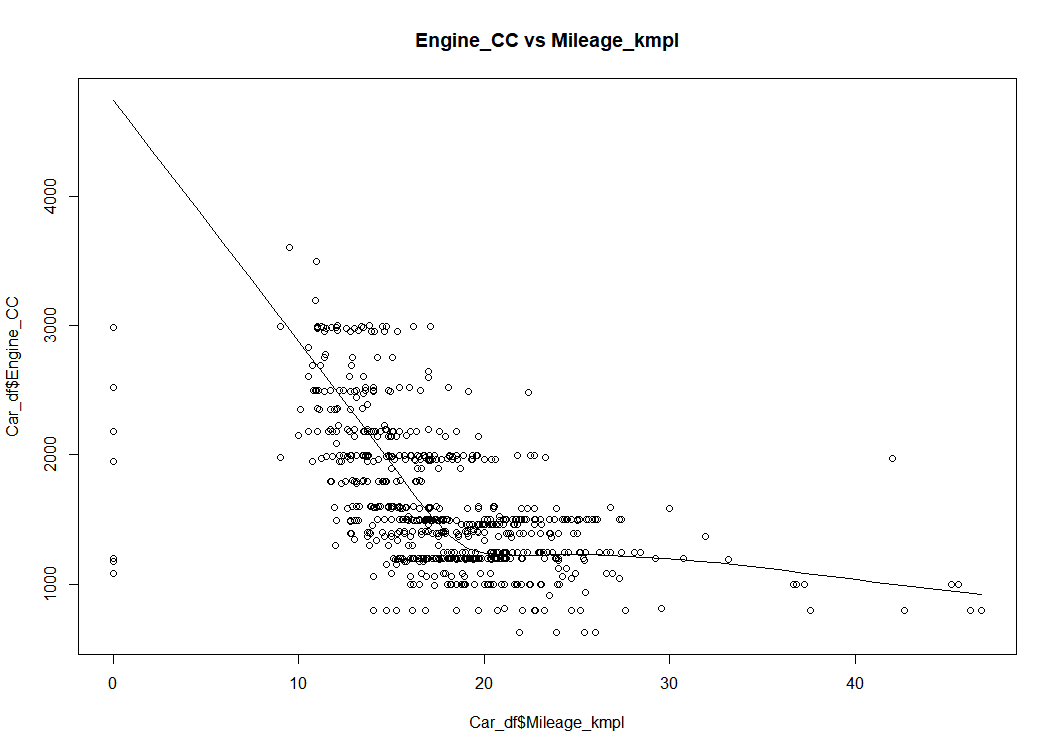
There are only two types of transmissions that are manual and automatic. According to the bar plot, we can easily conclude that most car brands are manual as compared to automatic.

4. Seller Type



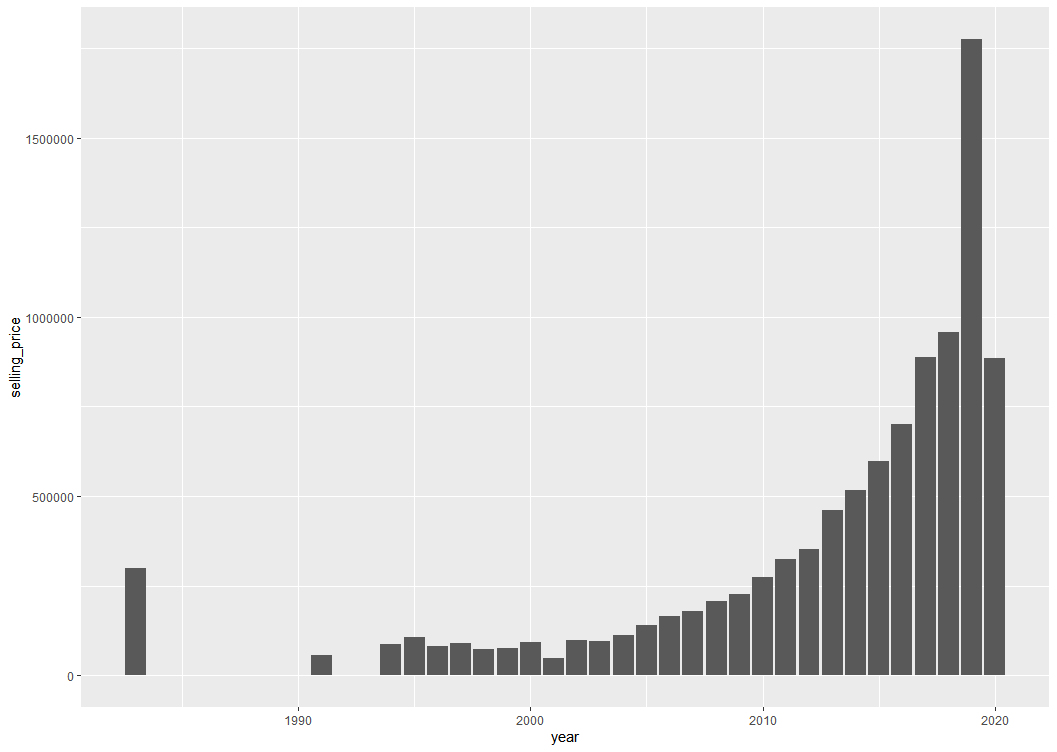
According to data, there are only three types of sellers in the market i.e., individual, dealer, and trustmark dealer. According to the bar plot, we can easily conclude that most seller types are individual as compared to others.

Mileage (kmpl) vs Engine (CC)



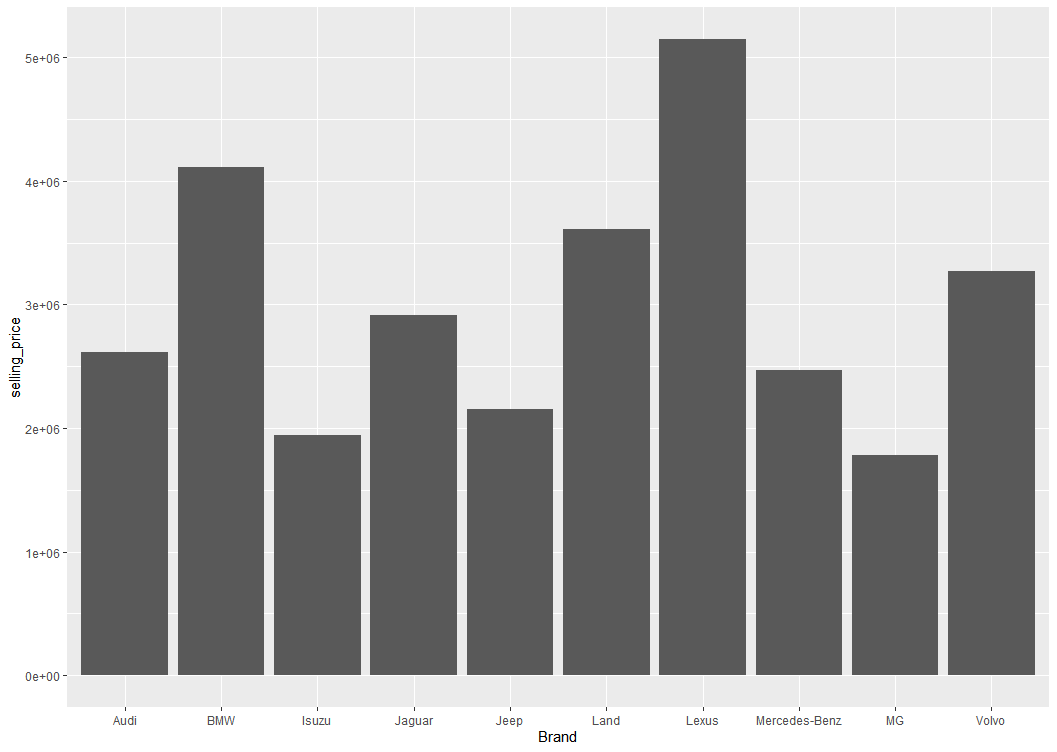
The above diagram shows the scatter plot between the engine (CC) and mileage of the cars. The scatter plot is more appropriate for these variables because its data points are continuous. It shows a negative relationship between the engine (cc) and mileage of the cars for the reason that plot points seem to form a line going down from left to right. In general, the larger the engine (CC's), the lower mileage it will achieve.

Yearly Average Selling Price of Cars



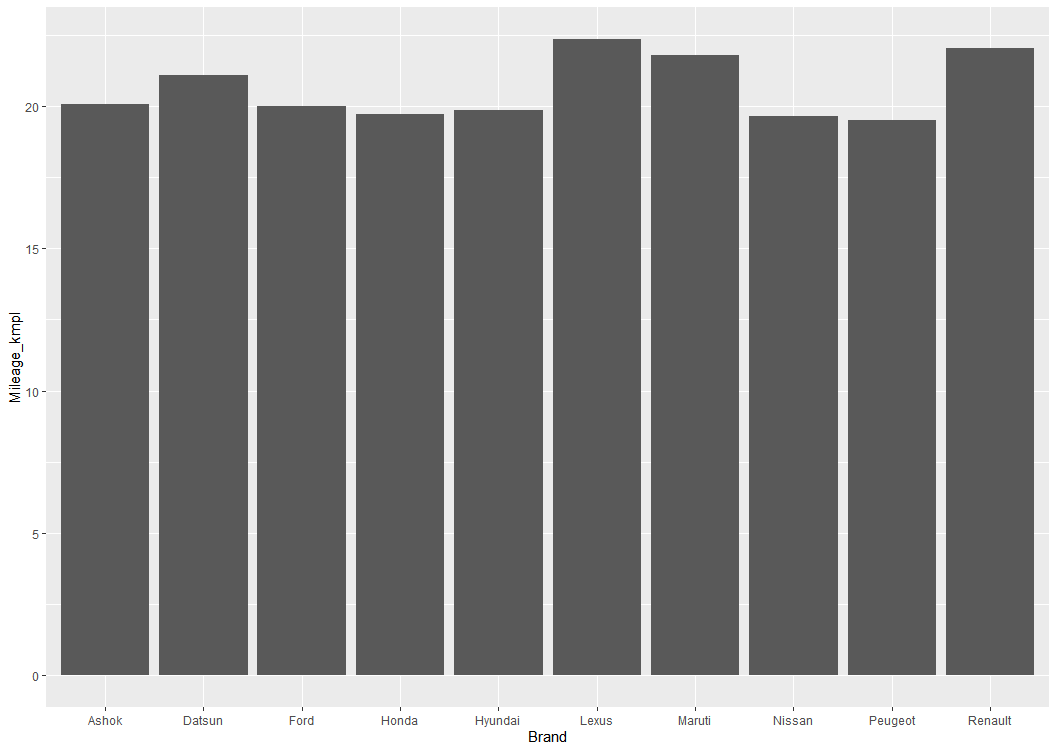
As we performed in part 1 by using MS Excel. Thus, the result is the same. Therefore, we can use the same explanation as it is.

Top 10 Car Brands



As we performed in part 1 by using MS Excel. Thus, the result is the same. Therefore, we can use the same explanation as it is.

Top 10 Brand Mileage



As we performed in part 1 by using MS Excel. Thus, again the result is the same. Therefore, we can use the same explanation as it is.

Part 3------------------------Linear Regression-------------------------------------

R Code:

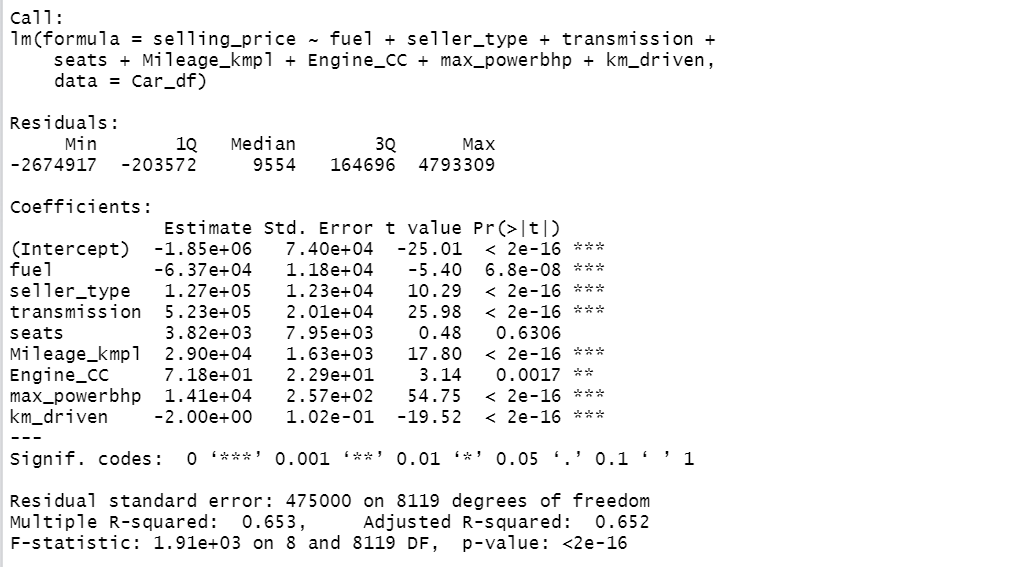
y\_var <- data.frame(Car\_df$selling\_price)

model\_reg <- lm(formula = y\_var ~ fuel+seller\_type+transmission+

seats+Mileage\_kmpl+Engine\_CC+max\_powerbhp+km\_driven, data = Car\_df)

summary(model\_reg)

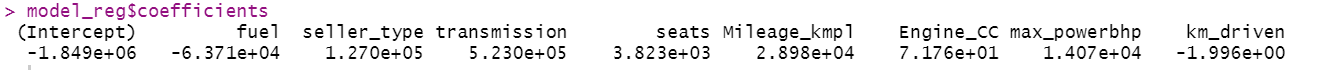
Output of linear regression model:



Regression Equation:

The dependent variable in this model is the selling price of cars and others are the independent variables but we take only relevant variables.

The equation shown below is regression equation.



If slope coefficient is positive which means that the relationship between the dependent variable and the independent variable is positive, whereas, if the slope is negative, then the relationship between the dependent variable and the independent variable is negative. For example, we pick only one variable for interpretation, the slope of mileage is 29,000, implying that for each increase of 1 kmpl in mileage, then the value of selling price is estimated to increase by 29000.

Model Diagnosis:

The R-square of the model is 65%. The interpretation is that 16% of the variation in the selling price (or dependent variable) can be explained by the variation in the independent variables.

The 65% of R-square in the model indicates that the regression model of the data is satisfactory.

The p-value of all the independent variables is less than zero except seats. So, the p-value of all the independent variables is statistically significant except seats for the reason that the p-value is less than the alpha level that is 0.000 < 0.05.

On the other hand, the p-value of the overall model is less than zero which indicates that the model is statistically significant.

Appendix (R Code)

library(readr)

library(dplyr)

library(tidyverse)

library(lmtest)

library(olsrr)

setwd("D:/Augment Systems Kasotiya/Excel/15. Car Dataset Excel and R- AH078063BB")

# Import Data set (Car)

Car\_df <- read.csv("Car details v3.csv")

View(Car\_df)

library(psych)

describe(Car\_df)

# Looking at null values

Car\_df %>%

summarise\_all(~ sum(is.na(.)))

# ---------------- Data preprocessing-------------------

## For name variable

split\_cars <- strsplit((Car\_df$name), " ")

split\_cars

Brand <- sapply(split\_cars, "[",1)

Brand

Car\_df['Brand'] <- c(Brand)

#View(Car\_df)

Car\_df <- subset(Car\_df, select = -c(name)) #Drop name column

View(Car\_df)

## For mileage

#split\_mileage <- strsplit((Car\_df$mileage)," ")

#View(split\_mileage)

#Mileage\_kmpl <- sapply(split\_mileage, "[",1.2)

#View(Mileage\_kmpl)

#Mileage\_kmpl

#OR

Mileage\_kmpl <- parse\_number(Car\_df$mileage)

Mileage\_kmpl

Car\_df['Mileage\_kmpl'] <- c(Mileage\_kmpl)

View(Car\_df)

### Conversion of km/kg to kmpl

Car\_df <- Car\_df %>%

mutate(Status = case\_when(endsWith(mileage, "g") ~ 1,

endsWith(mileage, "l") ~ 0))

Car\_df <- Car\_df %>%

mutate(Mileage\_kmpl = if\_else(Status == 1, Mileage\_kmpl\*1.4, Mileage\_kmpl))

## For Engine (CC)

Engine\_CC <- parse\_number(Car\_df$engine)

Engine\_CC

Car\_df['Engine\_CC'] <- c(Engine\_CC)

View(Car\_df)

max\_powerbhp <- parse\_number(Car\_df$max\_power)

max\_powerbhp

Car\_df['max\_powerbhp'] <- c(max\_powerbhp)

View(Car\_df)

#Car\_df[4934,]

#Drop Old or unused Variables

Car\_df <- subset(Car\_df, select = -c(mileage, Status, engine, max\_power, torque)) #Drop columns

View(Car\_df)

# Conversion of Categorical to Numerical Value

str(Car\_df)

unique(Car\_df$fuel)

Car\_df <- Car\_df %>%

mutate(fuel = case\_when(fuel == "Diesel" ~ 1,

fuel == "Petrol" ~ 2,

fuel == "LPG" ~ 3,

fuel == "CNG" ~ 4,

TRUE ~ 0))

#unique(Car\_df$fuel)

#unique(Car\_df$seller\_type)

Car\_df <- Car\_df %>%

mutate(seller\_type = case\_when(seller\_type == "Individual" ~ 1,

seller\_type == "Dealer" ~ 2,

seller\_type == "Trustmark Dealer" ~ 3,

TRUE ~ 0))

unique(Car\_df$seller\_type)

unique(Car\_df$transmission)

Car\_df <- Car\_df %>%

mutate(transmission = case\_when(transmission == "Manual" ~ 1,

transmission == "Automatic" ~ 2,

TRUE ~ 0))

#unique(Car\_df$transmission)

unique(Car\_df$owner)

Car\_df <- Car\_df %>%

mutate(owner = case\_when(owner == "First Owner" ~ 1,

owner == "Second Owner" ~ 2,

owner == "Third Owner" ~ 3,

owner == "Fourth & Above Owner" ~ 4,

owner == "Test Drive Car" ~ 5,

TRUE ~ 0))

View(Car\_df)

# Filling Missing Values

#1.

Car\_df$Mileage\_kmpl[is.na(Car\_df$Mileage\_kmpl)]<- mean(Car\_df$Mileage\_kmpl, na.rm=TRUE)

View(Car\_df)

#2.

Car\_df$Engine\_CC[is.na(Car\_df$Engine\_CC)]<- median(Car\_df$Engine\_CC, na.rm=T)

View(Car\_df)

#3.

Car\_df$max\_powerbhp[is.na(Car\_df$max\_powerbhp)]<- mean(Car\_df$max\_powerbhp, na.rm=T)

View(Car\_df)

#4.

Car\_df$seats[is.na(Car\_df$seat)]<- median(Car\_df$seat, na.rm=T)

View(Car\_df)

#Check Correlation

#install.packages("ggcorrplot")

library(ggcorrplot)

df <- subset(Car\_df, select = -c(Brand)) #Drop name column

View(df)

corr <- round(cor(df), 1)

ggcorrplot(corr, hc.order = TRUE, type = "lower",

lab = TRUE)

## Graphical Analysis

library(ggplot2)

ggplot(Car\_df, aes(x = fuel)) +

geom\_bar()

ggplot(Car\_df, aes(x = owner)) +

geom\_bar()

ggplot(Car\_df, aes(x = transmission)) +

geom\_bar()

ggplot(Car\_df, aes(x = seller\_type)) +

geom\_bar()

scatter.smooth (x=Car\_df$Mileage\_kmpl, y=Car\_df$Engine\_CC,

main="Engine\_CC vs Mileage\_kmpl")

avg\_sp <- aggregate(selling\_price ~ year, Car\_df, mean)

avg\_sp

avg\_sp <- data.frame(avg\_sp)

avg\_sp

ggplot(avg\_sp)+

geom\_bar(mapping = aes(x = year, y = selling\_price), stat = "identity")

brand10 <- aggregate(selling\_price ~ Brand, Car\_df, mean)

brand10

data.frame(brand10)

brand10 <- brand10[order(brand10$selling\_price, decreasing = TRUE),][1:10,]

brand10

mileage10 <- aggregate(Mileage\_kmpl ~ Brand , Car\_df, mean)

mileage10

data.frame(mileage10)

mileage10 <- mileage10[order(mileage10$Mileage\_kmpl, decreasing = TRUE),][1:10,]

mileage10

data.frame(mileage10)

ggplot(mileage10)+

geom\_bar(mapping = aes(x = Brand, y = Mileage\_kmpl), stat = "identity")

#------------------------Linear Regression-------------------------------------

# x and y Determination

#x\_var <- data.frame(Car\_df$fuel, Car\_df$seller\_type, Car\_df$transmission, Car\_df$owner)

y\_var <- data.frame(Car\_df$selling\_price)

model\_reg <- lm(formula = y\_var ~ fuel+seller\_type+transmission+

seats+Mileage\_kmpl+Engine\_CC+max\_powerbhp+km\_driven, data = Car\_df)

print(summary(model\_reg))

model\_reg$coefficients